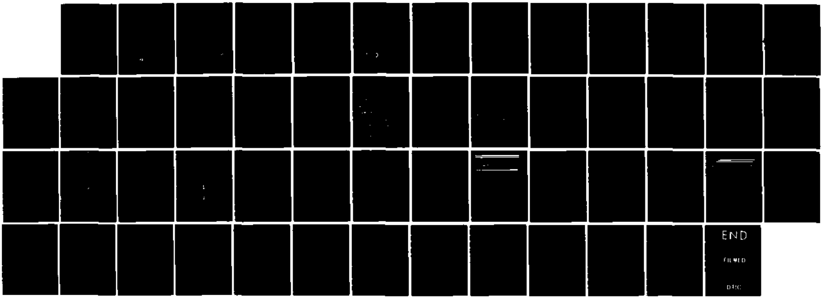
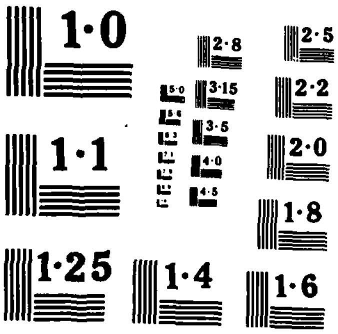
AD-A152 589 REPRESENTING DYNAMIC SKILL KNOWLEDGE(U) UNIVERSITY OF SOUTHERN CALIFORNIA REDONDO BEACH BEHAVIORAL TECHNOLOGY LABS A MUNRO ET AL. DEC 84 TR-103 N00014-83-K-0081 1/1

UNCLASSIFIED F/G 5/10 NL




1·0
2·8
2·5
5·0
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

2

REPRESENTING DYNAMIC SKILL KNOWLEDGE

Allen Munro
Douglas M. Towne
David S. Surmon

December, 1984

Technical Report No. 103

BEHAVIORAL TECHNOLOGY LABORATORIES
Department of Psychology
University of Southern California

Sponsored by
Personnel and Training Research Group
Office of Naval Research

Under Contract No. N00014-83-K-0081
ONR NR 667-506

REPRESENTING DYNAMIC SKILL KNOWLEDGE

Allen Munro
Douglas M. Towne
David S. Surmon

December, 1984

Technical Report No. 103

BEHAVIORAL TECHNOLOGY LABORATORIES
Department of Psychology
University of Southern California

Sponsored by
Personnel and Training Research Group
Office of Naval Research

Under Contract No. N00014-83-K-0081
ONR NR 667-506

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A152 589	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Representing Dynamic Skill Knowledge		5. TYPE OF REPORT & PERIOD COVERED Final Report 11-01-82 to 12-31-84
		6. PERFORMING ORG. REPORT NUMBER Technical Report 103
7. AUTHOR(s) Allen Munro, Douglas M. Towne and David S. Surmon		8. CONTRACT OR GRANT NUMBER(s) N00014-83-K-0081
9. PERFORMING ORGANIZATION NAME AND ADDRESS Behavioral Technology Laboratories University of Southern California 1845 S. Elena Ave., Redondo Beach, CA 90277		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 667-506
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 North Quincy Street Arlington, VA 22217		12. REPORT DATE December 1984
		13. NUMBER OF PAGES 36 + iv
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release: distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) knowledge representation, dynamic skill training, simulation training		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Intelligent Computer Aided Instruction (ICAI) requires machine understanding of student knowledge and skills. Methods for representing knowledge about a coherent set of tasks, dynamic skill tasks, are presented. A formal system appropriate to the representation of temporality, events, and actions is adopted. A semantic composition method naturally promotes a "degrees of specificity" approach to the representation of knowledge. Some, presumably widely held, elements of dynamic skill knowledge are domain-independent. Others are specific to particular task domains. Many task-specific representations can be seen as instantiations of more generic representations. Student explanations of a variety of dynamic skill tasks are analyzed in this framework. An experimental computer-based dynamic skill task is used to permit detailed observation of student performances that are analyzed in terms of the proposed representation format.

ABSTRACT

Intelligent Computer Aided Instruction (ICAI) requires machine understanding of student knowledge and skills. Methods for representing knowledge about a coherent set of tasks, dynamic skill tasks, are presented. A formal system appropriate to the representation of temporality, events, and actions is adopted. A semantic composition method naturally promotes a "degrees of specificity" approach to the representation of knowledge. Some, presumably widely held, elements of dynamic skill knowledge are domain-independent. Others are specific to particular task domains. Many task-specific representations can be seen as instantiations of more generic representations. Student explanations of a variety of dynamic skill tasks are analyzed in this framework. An experimental computer-based dynamic skill task is used to permit detailed observation of student performances that are analyzed in terms of the proposed representation format.

TABLE OF CONTENTS

LIST OF FIGURES

REPRESENTING DYNAMIC SKILL KNOWLEDGE

BACKGROUND

New hope for the future of computer-based instruction (CAI) comes from three sources. One is a trend toward software systems that do not appear to be CAI in the traditional sense. These are programs that present a rich, manipulable domain to learners. Experimental examples of such systems include Borning's (1979) ThingLab and the Equipment Simulation Authoring System (ESAS) (Towne & Munro, 1984). In the commercial sector, the Filevision program for the Apple Macintosh computer also has the characteristic of providing users with undirected access to a complex environment of data and graphics over which they have direct control.

Traditional CAI systems provide students with only minimal control over the systems they make use of. The new trend in interactive applications does not force the author of an application to envision every possible use that might be made of it. In CAI systems such as PLATO or TICCIT, every branch that a student might take must be explicitly coded by the application author. In ThingLab, ESAS, and Filevision, authors create rule-governed environments that users can explore in a free-play fashion. This trend, if pursued, will liberate CAI from the Programmed Instruction paradigm. Students will become tool-users, able to use computer systems as devices to aid learning, just as writers use word processing programs as an aid to writing and financial analysts use spreadsheets as an aid to decision making.

A second hopeful trend in CAI, exhibited by the same systems cited as examples of student-controlled applications, is that of providing application authors with object-oriented methods for building applications. Authors can select meaningful objects and compose them into simulated, functioning scenes using these systems. Because computer programming skills are not required to make use of such authoring systems, subject matter experts can produce high-quality simulations at lower cost.

The most important next stage advances in CAI are likely to come in the area of intelligent instruction -- the result of applying artificial intelligence methods to training problems. This is the third trend that is likely to have significant effects on CAI, increasing its effectiveness and extending the domains in which it can be effectively and economically employed. We are particularly interested in the problem of providing supervisorial intelligence for dynamic simulation training. An intelligent tutor could provide individualized coaching, instruction, and simulation control in response to the course of events in training simulations.

Two basic issues must be resolved if intelligent tutorial is to be successfully implemented for simulation training. The first is to determine a combination of representation format and methods of analysis that can be used to represent expert and student understanding of the task for the purposes of training. The second is to provide a method by which a subject matter expert can encode the expertise required by the tutor -- in effect, to provide an authoring system for intelligent tutors.

This report deals with the first of these issues -- devising a representation for generic dynamic skill knowledge. Based, in part, on a format devised by Allen (1984), the representation includes semantic primitives appropriate for the representation of elementary concepts of dynamic skill tasks, including duration, simultaneity, temporal overlap, and do on. An intermediate level of representation, composed of the primitive elements, conveys a number of dynamic skill concepts such as Wait-For-Event-to-Perform, and so on. At a higher level, it provides a generic schema for interactive task environments.

REPRESENTATION FUNDAMENTALS

One objective of our work has been to begin assembling a general inventory of characteristics of dynamic tasks. By examining a set of specific task environments, in a manner similar to the exploratory work of Kieras concerning device representations (Kieras, 1984; Kieras & Bovarir, 1983), we have identified a significant number of processes, actions, and events required to represent a range of dynamic tasks.

The following six rule forms or rule schemata are typical of the primitive elements which are involved in many dynamic tasks:

1. Action a must precede action b.
2. Action a must be performed immediately before action b.
3. Action a must be started before event b occurs.
4. Action a must be finished before event b occurs.
5. Action a must be initiated within <t time-units> following event b.
6. Action a must not be performed during event b.

At a somewhat less abstract level, many dynamic task representations must refer to elementary concepts from a small set of positional and movement primitives. These can be expressed by statements such as

7. Object i is near object j.
8. Object i is below object j.
9. Object i is heading toward object j (or, in direction d).
10. Object i has speed s.
11. Object i is new.
12. Object i has disappeared.

Dynamic skill tasks appear to require a semantic inventory that includes such concepts as precede, started before, finished before, not during, near, below, heading toward, speed, and so forth. Many more complex dynamic skill concepts can be defined in terms of such more primitive elements.

The primary rationale for developing this inventory was to determine what form of representation system is required to accomodate a wide range of complex time-constrained man-machine interactions. Rules 1 and 2, for example, express simple sequence constraints of the type that would be used to represent a serial-action task (a non-dynamic task in which durations are irrelevant, such as equipment troubleshooting). In general, serial-action task environments may be treated as special cases of the more general dynamic task environment. Moreover, a dynamic task may have aspects which are serial in nature. Thus rules 1 and 2 are useful for characterizing some portions of dynamic tasks.

Rules 3, 4, and 5 reflect some of the nature of dynamic tasks, in which the performance of one action is related to the timing of other actions and events. Rule 3 also allows a degree of simultaneaty, while rule 6 expresses a specific prohibition against the concurrency of events a and b.

While the generic primitives for dynamic tasks are expressed in English, they provide a clear indication of the flexibility and range of relationships needed in a formal representation system. The six temporal relationship primitives listed above demonstrate three of the central requirements which a computational theory of action must meet:

1. Effects and interactions among actions must be expressible and computable; it is not sufficient, for example, to simply represent temporal order of events or simple causal relationships with primitive logical operators of the form "if a then b".

2. Simultaneous actions and events must be expressible, with no inherent limitation on the number of such concurrent elements. The situation calculus (McCarthy & Hayes, 1969), which represents the world with static states acheived via the performance of particular actions, does not satisfy this need.

3. The formalism must be able to express inactivity, as in rule 6, and non-continuous activity (such as resumming an interrupted action).

Other requirements of a formalism result from its intended basis for instructional purposes. These include the need to recognize a performer's intentions and plans and the need to represent time in a manner which would allow tutorial functions to recognize time-critical situations. (For example, if a tutor observes a student strike a function key just a fraction of a second too late after performing a number of equally critical actions within their alloted time, it may be inappropriate to classify the error as "forgetting to type the function key during the required period.")

A representation system which adequately accomodates a useful range of dynamic task environments must be extremely rich, flexible, and adaptable. We believe this can be best accomplished by constructing a relatively small number (less than one hundred, perhaps) of generic primitive rules. From these, a second tier of more specific rules would be constructed. At this (second) level or above, rules would be constructed to meet particular needs of an application, such as the Air Intercept Controller (AIC) task. We would expect that many of the second-and third-tier rules to also be of general use in future applications. Each domain-specific rule would be expressed in terms of lower-level domain-independent elements. Possibly some would be built using the first-tier primitives, others would involve more complex construction.

Dynamic Task Primitives

We propose to construct formal dynamic task primitives to represent the English rules in our current inventory using the formalism for reasoning about actions developed by J. F. Allen (AI, 1984). This formalism is based upon a theory of temporal logic which recognizes time intervals rather than distinct points. Allen defines a small set of predicates which express the properties of various time periods. A fundamental predicate is HOLD, described below:

HOLD (p,t)   which is true if and only if property p holds during time t

and a number of predicates related to time periods:

DURING (t1,t2)   time interval t1 is fully contained in t2

STARTS (t1,t2)   time interval t1 shares the same beginning as t2, but ends before t2 does.

FINISHES (t1,t2)   time interval t1 shares the same end as t2, but begins after t2 begins.

BEFORE (t1,t2)   time interval t1 is before interval t2, and they do not overlap in any way.

OVERLAP (t1,t2)   time interval t1 starts before t2, and they overlap

MEETS (t1,t2)   interval t1 is before interval t2, but there is no interval between them; t1 ends where t2 starts.

EQUAL (t1,t2)   t1 and t2 are the same interval.

-- Allen (1984)

From these, higher level predicates may be constructed. For example IN(t1,t2) can be defined in terms of DURING, START, and FINISHES to express that t1 is wholly contained in t2.

HOLD specifies the truth of a property over a time period. A second key object in the ontology, OCCUR(e,t), is true if and only if event e <u>happened</u> during time t (and not during any subinterval in t).

In most respects, the primitive predicates have the same role in more than they describe, is supported by student responses to informal questions about described tasks, by their performance on the tasks, and by interpretations of their sometimes sparse descriptions that impose consistency.

To analyze individual student's representations for tasks, methods must be devised that do not rely on verbal or textual descriptions. A potentially ideal method would make use of student task performance to derive such representations.

propositions from a composite representation, can be developed for eliminating obvious errors from the "inclusive correct" representation. The result of applying this procedure is to construct a model of an ideal all-knowing task expert. This expert model includes all the correct information displayed in the collected descriptions of the task, while excluding individual errors.

The approach of propositionally analyzing many descriptions of a task may prove to be a fruitful one for deriving "possible expert" knowledge about a task domain -- the knowledge that some deal expert might have about the task. This approach has less appeal for determining appropriate individual representations of task knowledge. A method that purports to discover an individual's knowledge about a task must make use of task performance data.

Not all elements of a representation such as the composite propositional representation for the PLAY-INVADERS task (given on page 10 and in a notational variant in Figure 1) are equally central to an expert's conception of the task. For example, the ideas of shooting the aliens and avoiding their weapons are much more important than the concept that the aliens' weapons destroy the blockades. Frequency of mention is one measure of centrality of component concepts.

Propositional analysis of many task descriptions may be a useful method for deriving a comprehensive "expert" representation of the task, but it is not, by itself, an adequate method for determining individual representations. Individual task descriptions can reveal intrusive errors in task understanding. They cannot be relied on to reveal gaps in understanding, however, because many students fail to describe much that they seem to know about the task. This claim, that students know

A composite representation (either graphical, as in Figure 1, or propositional, as at the beginning of this section) can be constructed on the basis of the descriptions of many subjects. Each student's coded propositions can be related to the composite representation, largely by identifying identities of reference, but sometimes by adding new propositions.

What is the status of a composite representation derived by this method? One claim that can be made about such representations is that they are fictions. According to this claim, no single student knows everything about the task that is conveyed in the composite representation. Since there is no "group mind" that holds the entire composite, it does not make sense as a representation. The only representations that make sense are representations for individuals' understandings, based on their descriptions.

A second claim is that the composite representation is a good representation for most students' understandings of the task. According to this claim, all the students have this essential understanding of the task, but some are better at expressing it than are others.

Neither claim can be completely correct. Many students seem to know more about the task than they include in their descriptions. A representation such as that given by Student #3 lacks the cohesiveness required for a complete task description. Furthermore, we suspect that many of the students would perform the task in a much more adequate fashion than their descriptions would suggest. On the other hand, some students have incomplete and/or incorrect ideas about the task. We certainly would not want to include erroneous propositions in the composite representation. It would not make much sense to say that a student shared in the composite representation when performance data suggested clear errors and omissions.

Operational methods, such as excluding all idiosyncratic

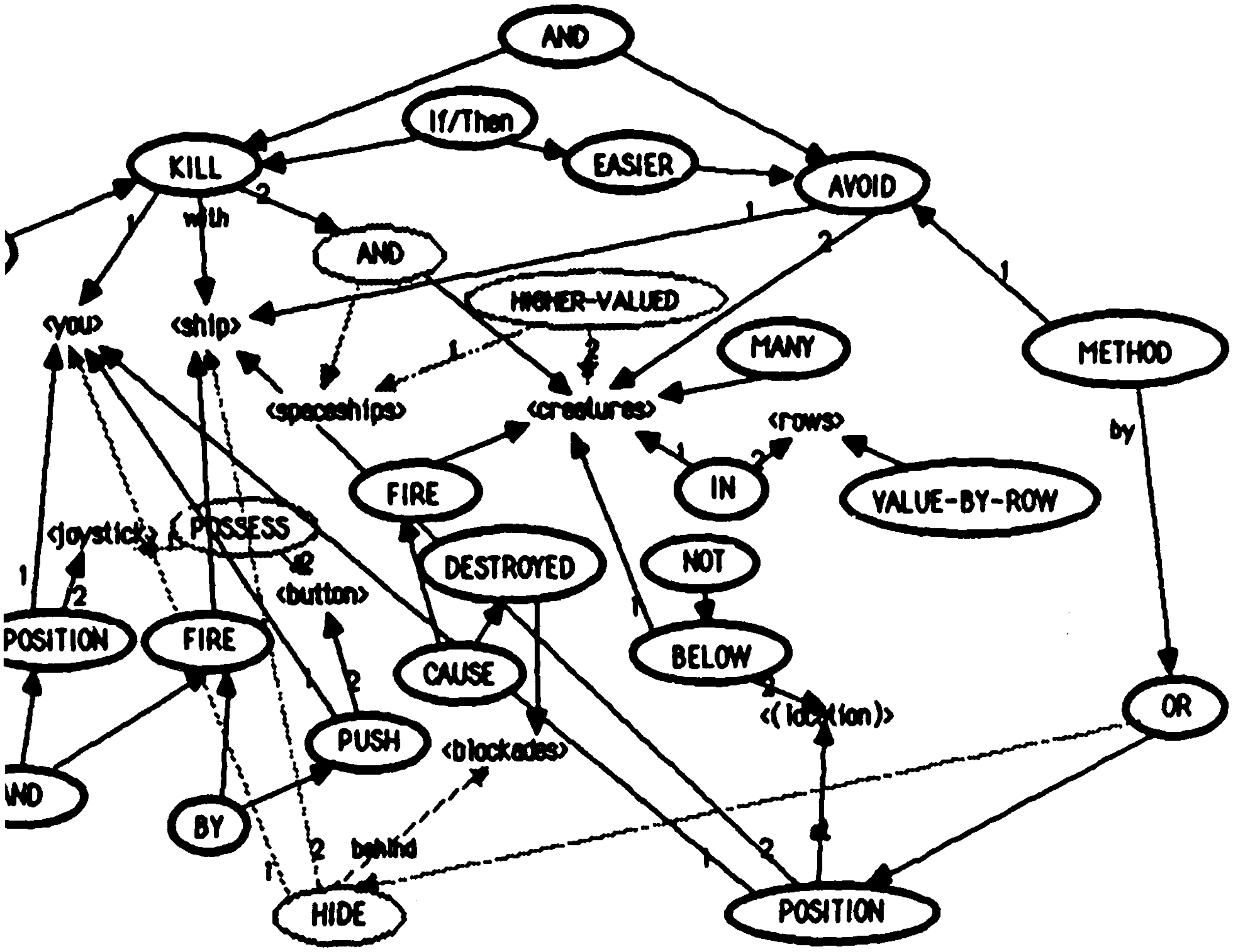


Figure 4.

Subject 1

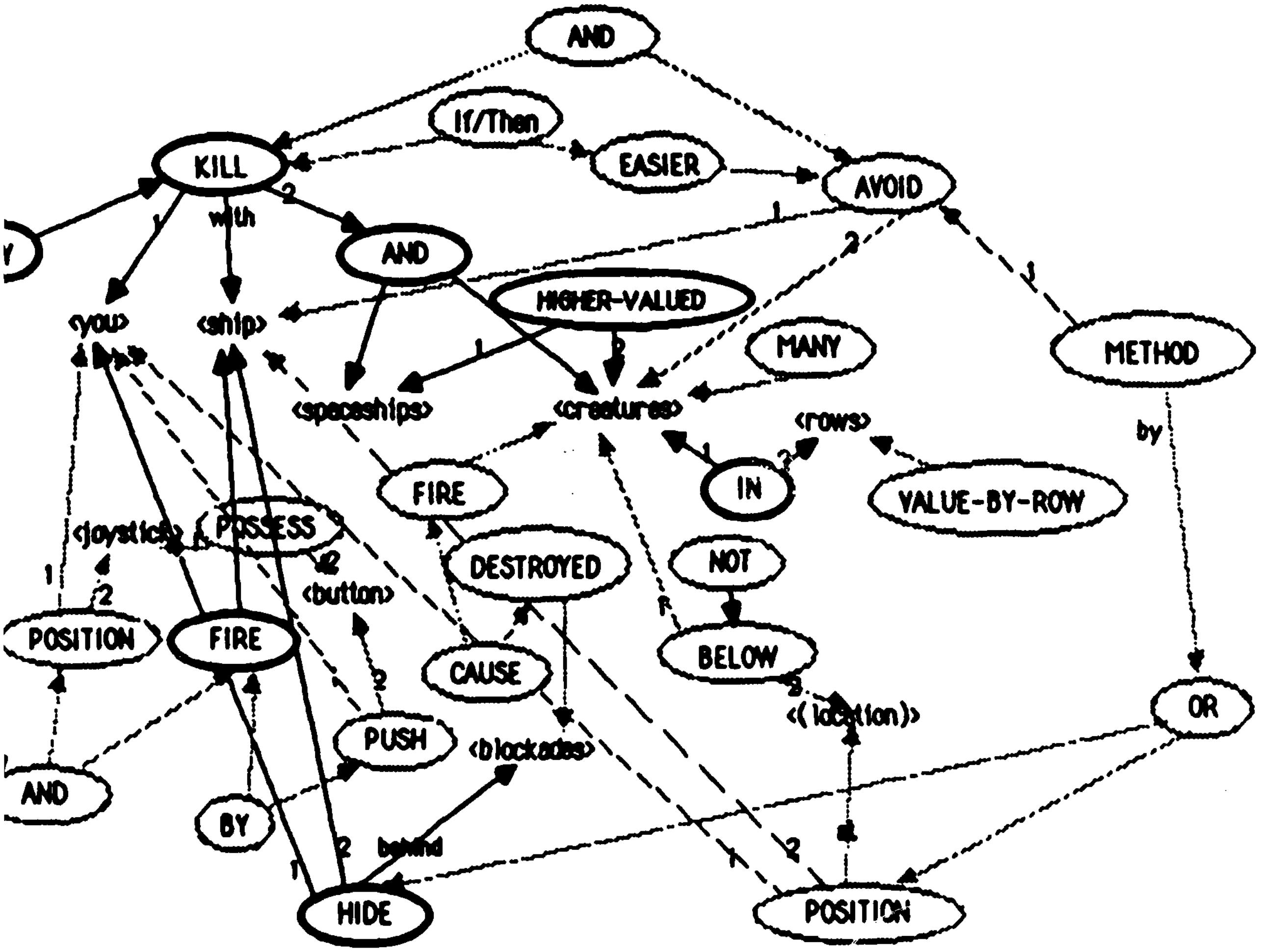


Figure 3.

Subject 25

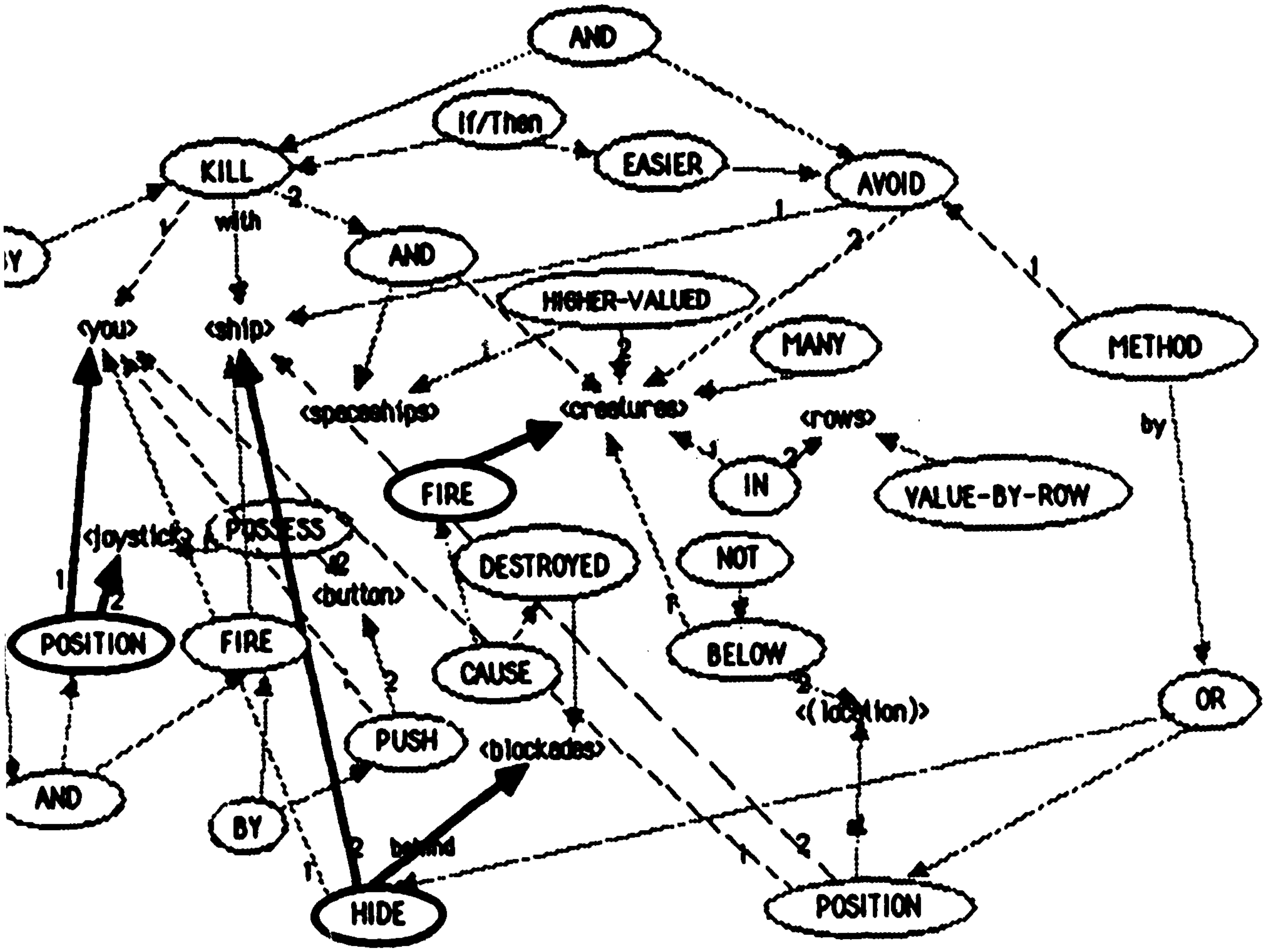


Figure 2.

Subject 3

The relationship between a particular student's description of a the task and the composite representation can be more easily appreciated when both are presented as network graphs. Figure 1, "Composite Semantic Reperesentation" conveys the same information included in the sequence of propositions displayed at the beginning of this section. Particular instantiations (or partial instantiations) of predicates are represented as ovals surrounding the predicate label. Subscripts are unnecessary, since identity of reference is portrayed by pointing to the same oval.

The pointers represent the bindings between predicate instantiations and their arguments (which may themselves be predicate instantiations). The argument bindings (arrows) may be labeled to reflect the roles of the arguments with respect to the predicate. Labels can be prepositions such as "at", "by", and so on, or they can be numbers, where "1" represents an agentive argument, "2" an objective argument, "3" a patient argument, and so on (Fillmore, 1968). The nominal arguments are represented by the name of the type of the object, enclosed in angled brackets.

Individual student descriptions can be translated from the propositional representation to this semantic network representation. Figure 2 presents a semantic network representation for student #3. It presents the same set of propositions given above for this student, shown against a "grayed out" background of the composite representation. The same method can be used to display the content of task descriptions given by other students, including those students who describe the task much more fully. Figure 3 presents the content of the description of the PLAY-INVADERS task given by subject 25, who described the task in moderate detail. Figure 4 presents the semantic network representation of the description given by Subject 1, which includes most of the propositions in the composite representation.

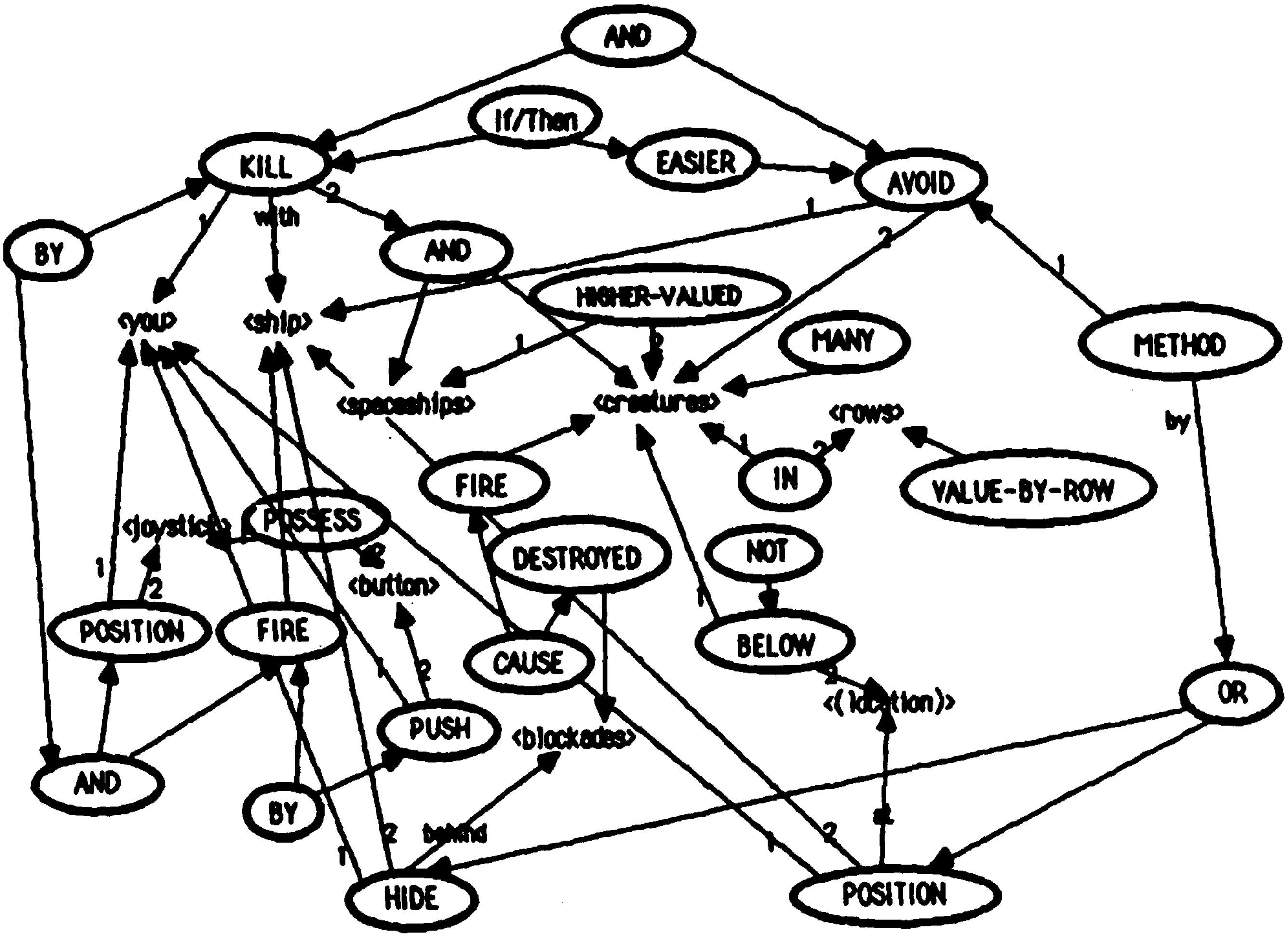


Figure 1.

Composite Semantic Representation

```
POSITION (agent, object, {instrument, source, goal})  =
  COMPOSITE
     ($DO_i$ (agent, PRO-ACTION (agent, instrument)),
     CAUSE ($DO_i$ (...), $CHANGE\text{-}POS_j$ (object, source, goal)))
```

Invoking this definition in the context of $POSITION_y$ (...) results in an expansion based on the definitions of POSITION and of PLAY-INVADERS.

```
$POSITION_y$ (agent, <ship>, at <(location)>)  ==>
  $COMPOSITE_y$
     ($DO_{ii}$ (agent, PRO-ACTION (agent, instrument)),
     CAUSE ($DO_{ii}$ (...), $CHANGE\text{-}POS_{jj}$ (<ship>, , <location>)))
     $NOT_z$ ($BELOW_a$ (creatures, <location>)))
```

Even this representation is incomplete in that it does not specify that the PRO-ACTION is an instance of POSITIONing the joystick lever.

Representing the Task Descriptions of Individuals

Students differed a great deal in the amount of detail they included in their descriptions of the tasks. Some students gave very detailed and complete descriptions of the entire task, while others described only portions of the task in a few words. For example, student #3 said

> There are these different figures, invaders. They're firing down on the blockades. Your little guy hides behind them. You move the joystick.

Translating this description into the representation format used here produces a definition for PLAY-INVADER something like this:

```
PLAY-INVADER (agent)
  COMPOSITE
     ($POSITIONS_g$ (agent, <joystick>),
     $HIDE_x$ (agent, <ship>, behind <blockades>),
     $FIRE_f$ (<creatues>, <blockades>))
```

This representation makes use of a number of abbreviations. The nominal elements, shown here as nouns enclosed in angled brackets (<...>), have a more complex underlying representation. In the actual representation, the entities are represented as empty slots or memory locations about which predications such as SPACESHIPS(X_{4123}) or JOYSTICK(X_{4124}) are made. This treatment of objects permits a greater degree of uniformity in the underlying representation, since it is essentially identical to the treatment of actions and processes.

In ordinary language, what the above definition of PLAY_INVADERS says is that this task is a composite of many actions, processes, and states. The agent of PLAY_INVADERS must kill spaceships and creatures with a ship, while making the ship avoid the creatures. If the agent kills creatures, they become easier to avoid. The way to kill the creatures is to position the ship with a joystick and then fire by pushing a button that belongs to the joystick. More points are awarded for the spaceships than for the creatures. There are a lot of the creatures. The way to avoid the creatures is either to hid the ship behind the blockades or to move the ship to a location that is not below any creatures. The creatures are arranged in rows and the number of points scored for them varies with their row. The creatures fire at and destroy the blockades.

This representation is not at the level of primitives such as those described in the previous section. Each of the referenced predicates in the definition has its own definition, which can be invoked to further specify the meaning of PLAY_INVADERS. (In fact, some few of the above predicates are invocations of primitives, such as the instantiations of the primitive predicate AND.) For example, consider the partial instantiation of POSITION in the predication

POSITION_y (agent, <ship>, at <(location)>)

in the definition of PLAY_INVADERS. The generic definition of POSITION is devised for any action-oriented task. Based on student descriptions, here is a partial representation of a computer game similar to the arcade game "Invaders". This representation is a relatively complete one, and includes the major knowledge components that a moderately expert game player would have.

```
PLAY_INVADERS (agent) =
  COMPOSITE
     (AND$_h$ (KILL$_i$ (agent, AND$_j$ (<spaceships>, <creatures>), with <ship>),
          AVOID$_k$ (<ship>, <creatures>)),
     IF-THEN$_m$ (KILL$_i$ (...), EASIER$_n$(AVOID$_k$ (...))),
     BY$_o$ (KILL$_i$ (...), AND$_p$
          (POSITION$_q$ (agent, <joystick>), FIRE$_r$ (agent, <ship>)),
     BY$_s$ (FIRE$_r$ (...), PUSH$_t$ (agent, <button>)),
     HIGHER-VALUED$_u$ (<spaceships>, <creatures>),
     MANY$_v$ (<creatures>),
     METHOD$_w$ (AVOID$_k$ (...), OR (HIDE$_x$ (agent, <ship>, behind <blockades>),
          (POSITION$_y$ (agent, <ship>, at <(location)>))),
     NOT$_z$ (BELOW$_a$ (creatures, <(location)>)),
     VALUE-BY-ROW$_b$ (<rows>),
     IN$_c$ (<creatures>, (<rows>),
     POSSESS$_d$ (<joystick>, <button>),
     CAUSE$_e$ (FIRE$_f$ (<creatures>), DESTROY$_g$ (<blockades>)))
```

Composite Semantic Representation -- PLAY-INVADERS

The above representation uses subscripts to make references clear. For example, the KILL$_i$ (...) in the line that begins IF-THEN$_m$ refers to the same action that is given above as

(KILL$_i$ (agent, AND$_j$ (<spaceships>, <creatures>), with <ship>).

Co-referentiality of nominal entities is designated by name alone in this description. Subscripts are not necessary in these cases, since every instance of <spaceships> in this definition, for example, is meant to refer to the same set of entities.

DESCRIPTIONS OF DYNAMIC SKILL TASKS

One of the central problems in studying people's knowledge about dynamic tasks is determining what the state of that knowledge is. One method that can be employed is to ask a person to describe what he or she knows about the task, and to to translate what is said into a semantic representation, such as those given above. To the extent that their report reflects knowledge of the task, the representation will model that knowledge.

In order to explore this approach to the representation of task understanding, an observational procedure was conducted. Twenty-seven undergraduate students from the University of Southern California were presented with a series of five videotaped exemplars of dynamic skill tasks. Each taped segment was from two to three minutes in length and showed a dynamic task in progress. The students watched the taped segment and then briefly described the task, referring to objects displayed on the screen as necessary. Their descriptions were videotaped using a second video recorder, and later transcribed. These transcriptions could then be scored in a manner similar to that employed by Kintsch (1974).

The five taped segments displayed five tasks in action. One segment showed a personal computer game based on the "Invaders" arcade game. The screen showed a pair of hands manipulating a joystick that controlled the play of the game. Two other segments showed other joystick-controlled video games, one of the "Pac Man" type, and one based on "Asteroids." The fourth segment showed a version of the simulated Air Intercept Controller task, described in the following section. The fourth segment showed a pair of hands playing with a marble maze box, a dynamic skill task that does not require the use of a computer or video game.

Following Allen's representational format, a representation can be of a "screen pointer." Such lower-level instantiations constitute specific parameters for a generic schema at instantiation time.

In the generic version of a schema, parameters may be bound by selectional restrictions. For example, there may be a generic concept for "moving a screen pointer" which can apply to an event only if the object parameter is a screen pointer. Selectional restrictions limit the range of instantiations that can fill a parameter slot in a generic schema, but they are much less specific than actual instantiations.

The process of instantiation is often combined with a process of schema decomposition or expansion. At the time that a predicate is instantiated, some or all of the elements of its definition may also be instantiated. These instantiated predicates may themselves have definitional components that can further be instantiated, and so on. This expansion process is not a necessary part of every instantiation of a schema. Ordinarily expansion or decomposition will not take place or will be quite shallow. The depth of decomposition during instantiation reflects the depth of processing that the concept receives when it is activated.

Decomposition can be suspended during the instantiation process and later reactivated, due to further processing on the topic. Probable occasions for additional schema expansion include the instantiation of new elements that have parameters that are already instantiated, but not yet expanded. The additional activation due to the new "calling" schema may be enough to prompt the expansion of such a previously instantiated predicate.

Another primitive process that can be applied to all representation elements is comparison. Comparison processes must be able to determine identity, non-identity, function equivalence (equality), and inequality. Related processes are responsible for searching for particular representation elements or complexes of elements.

a representation that the more complex predicates that are constructed from them do. However, there are two respects in which the primitives are distinct. First, they cannot be expanded or "decomposed" into a more atomistic representation, as can complex predicates or schemata. Second, they have a special status in that a computer-based system that makes use of this representation must have pre-defined processes that are prepared to deal directly with the primitives, though not with more complex elements. For example, primitive truth-value operators can check the bounds of two time intervals to determine whether the *DURING* predicate applies.

Primitive Processes and the Representation

The most basic process that acts on the data that constitute a representation is instantiation. This is the process that makes a specific, context-embedded, and particularized copy of a generic schema. This copy represents a record of a portion of the mental activity accompanying a decision and its subsequent action. For example, the generic concept of "moving an object" must be distinguished from a specific memory for "moving a pointer on a screen from the lower left corner to the upper right corner at 11:42 A.M. on November 23." The latter concept is a specific instantiation of the former.

An instantiation or specific copy differs from the generic original in several respects. An instantiated copy of a predicate is part of a network of linked instantiations. For example, the instantiation for moving the screen pointer on November 23 may be linked to an instantiation for the appearance of a new icon on the screen immediately before. (This is especially likely if the pointer movement was a response to the new icon appearance.) A given instantiation has other instantiations linked to it as parameters, and it is typically a parameter for some higher-level instantiation of a different generic schema. The "object" parameter in this particular case of "moving an object" is an instantiation of the genric concept

COMPUTER-GENERATED REPRESENTATIONS OF STUDENT KNOWLEDGE BASED ON TASK PERFORMANCE

Ideally, an effective student knowledge representation construction system will build representations based on performance on a task. To explore the feasibility of such representation construction, it is necessary to use a dynamic skill task that the computer can monitor closely. For this purpose, we selected portions of a simulated Air Intercept Controller (AIC) task, exploiting our previous experience in developing experimental simulation training systems in this domain (Munro, Towne, & Fehling, 1981; Munro, Fehling, Blais, & Towne, 1981; Munro, Towne, Cody, & Abramowski, 1982; Munro, Fehling, & Towne, in press).

The AIC Tracking Task

In our previous research on representing dynamic task knowledge and skills, we have used a task simulation called the AIC tracking task. The student uses a pointing device (joystick, trackball, or mouse) to position symbols on blips that appear on a simulated radar screen. When a blip appears on the screen (after a radar sweep), the student puts a symbol on the blip as soon as possible. In Figure 5A, the student has labeled a new blip with the symbol "1".

In the AIC tracking task, a simulated tracking computer moves the symbols on a path established by the student. Ideally, this path should track the movements of the blips associated with the symbols. In order to establish a correct track, students must position the symbol a second time, after a second radar sweep. The two locations are used by the tracking computer to establish a path for the symbol to follow on subsequent updates. In 5B, the student has labeled the blip again in the new location. In 5C, the blip has changed course, and the student has relabeled it once again in order to establish a new correct path for the symbol ("1").

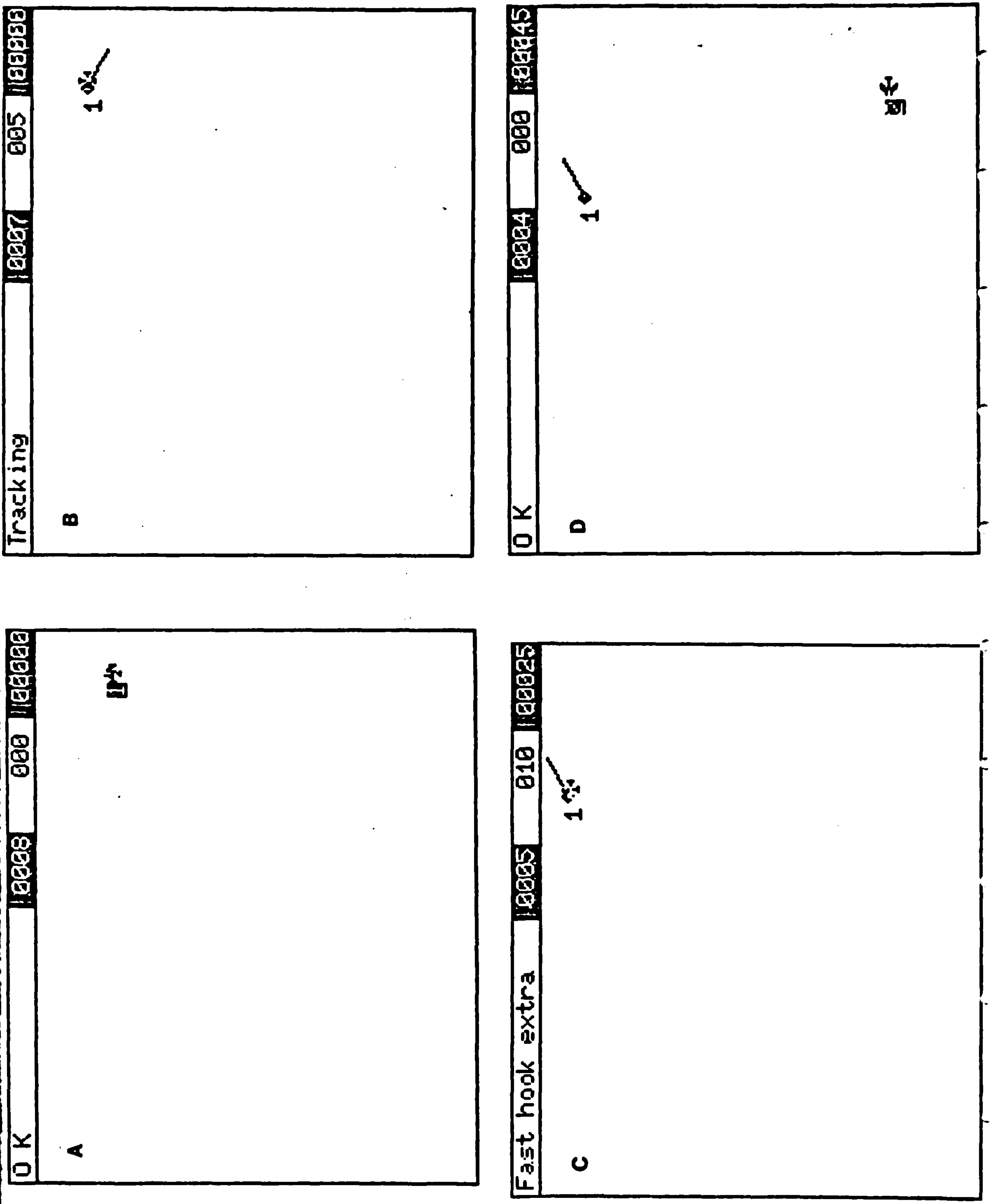


Figure 5.
Page 22

Students must observe and respond to a number of events that can occur during radar sweeps in this task. In Figure 6A, a new blip has appeared and been labeled before the next radar update. In 6B, the blip has been labeled again, thereby establishing a new correct path. In Figure 6C, however, the blip has changed velocity, so the computed tracking path is incorrect. The student must recognize this condition and relabel the blip again in order to keep the computed track on the course of the blip.

Formalizing AIC Tasks

From only the elements described above, many important elements of the AIC task can be represented formally. For example, one key (and surprisingly complex) event in AIC tracking is responding to a change of direction of a previously-identified blip (a "heading jink"). The change of direction characteristic would be represented with a function of the form

```
CHANGE-DIRECTION(BLIP,angle1,angle2)
```

To assert that a particular identified blip (BLIP1) changed direction during some particular time interval (T3), we would say

```
OCCUR(CHANGE-DIRECTION(BLIP1,DEGREES1,DEGREES2),T3)
```

and to determine if BLIP1 was previously identified we could evaluate the expression

```
OCCUR(IDENTIFIED(BLIP1,T2)) & BEFORE(T2,T3)
```

which states that BLIP1 was IDENTIFIED during interval T2, and T2 is BEFORE T3.

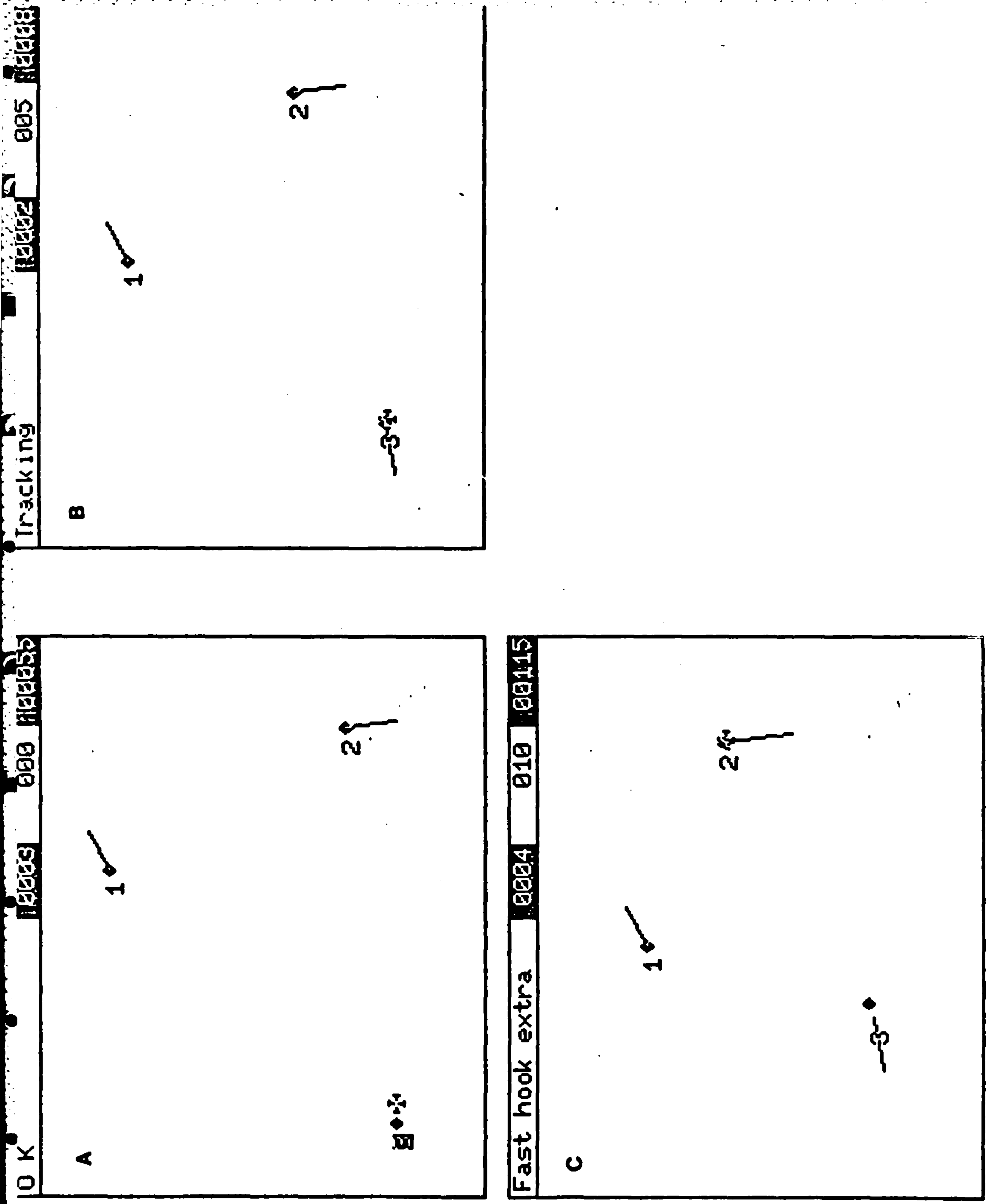

Tracking
0002
005
0008
B
1
2
3
OK
0003
000
00055
A
1
2
Fast hook extra
0004
010
00115
C
1
2
3

Figure 6

The proposed representation system will make use of decomposition methods in the course of deriving inferences. IDENTIFY has decomposable structure that, when expanded, makes explicit the role of an agent (the student) in an action that identifies a blip by assigning a symbol to it. To determine if BLIP1 was IDENTIFIED during T2, we would refer to the primitive definition of IDENTIFIED to ascertain whether it was satisfied DURING T2.

An important property of this representation system is that it can handle non-continuous processes, in addition to events and properties. Thus an intelligent tutorial system could recognize that a learner is 'tracking' a blip, even though that activity might be interrupted at times to take care of other time-critical needs. Of course it would be necessary to define explicitly what constitutes performance of the process. For example we might say that the process P, such as tracking, occurs over time interval T1 if there is no time interval T2, DURING T1 that is greater than T3 (a reference interval), in which the property of tracking is not true.

TRACKING(T1) is defined as

```
    NOT   T2 | OCCUR(NOT(ATTEND_TRACK),T2) & DURING(T2,T1) & T2>T3
```

where ATTEND_TRACK is defined in terms of student actions that reflect attention to the position of the blip.

If T3 were 20 seconds in duration, then any interruption in tracking greater than 20 seconds would terminate the time period during which tracking was being performed. If tracking resumed, another time period would be initiated.

Another universal aspect is the use of an "interactive situation skeleton" that is also built out of Allen's primitive building blocks. This semantic structure is a kind of universal schema for interactive dynamic skill tasks. One of the results of our current ONR research project has been the specification of this domain-independent expression of the top level structure of an interactive dynamic skill task. At the highest level of this representation, a dynamic skill consists of an agent causing processes in response to a task event, i.e.:

```
    OCCURRING(INITIATE(OBSERVE(agent,event),ACAUSE(agent,occurence)),t)
```

Here the ACAUSE clause represents the student's reaction to the observed event.

Tasks of interest, such as the AIC tracking task, have multiple objectives which must be continually and simultaneously considered by the performer. Thus the OCCURRENCE of AIC tracking is defined as a combination of positive objectives, and possibly the avoidance of negative events. Just how these competing factors are assigned levels of priority is a matter to be explored in future research.

When an intelligent tutor is created for use in a particular simulation training environment, most of the high-level representation components for that task will fit into slots in the interactive situation skeleton.

The system provides object-oriented representation of domain-specific knowledge for dynamic interactive tasks. Object-oriented composition systems have the advantages that they require less "intelligence" or expertise on the part of the user than do traditional simulation building methods (Boring, 1979; Hollan, 1983; Munro, 1984; Towne & Munro, 1984; Schneiderman, 1982). The user of this semantic object composition system is a process in the intelligent tutor, the process responsible for detecting evidence for particular representation components and for building student representations out of those components.

Implementation History

The original versions of the simulated AIC task developed at Behavioral Technology Laboratories were implemented in Apple Pascal on Apple // computers. In a new set of observational studies, a simplified version of the AIC task was created that emphasized a portion of the task, the "hooking" process, by which blips on the radar screen are labeled. The new version of the task (also implemented on Apple //s) was used to gather performance data. The purpose of studies with this task was to choose a subset of the AIC task that required dynamic skills but that was still amenable to complete representation.

Fifteen undergraduate students at the University of Southern California participated in this study for course credit. The students were presented with a brief videotape that introduced the AIC hooking task. Then they worked through a series of ten progressively more difficult practice exercises. These exercises were adaptive, so that the degree of difficulty depended on the skill with which a student performed the task. The students exhibited great variety in their performance. Some quickly attained near-expert performance levels, while others showed marked deficiencies in one or more aspects of the task. No single model of student understanding of task requirements and strategies could have accounted for all the different performances observed. Interviews with the students revealed that they had quite disparate models of the task and that, for particular students, aspects of performance could be related to the content of their descriptions.

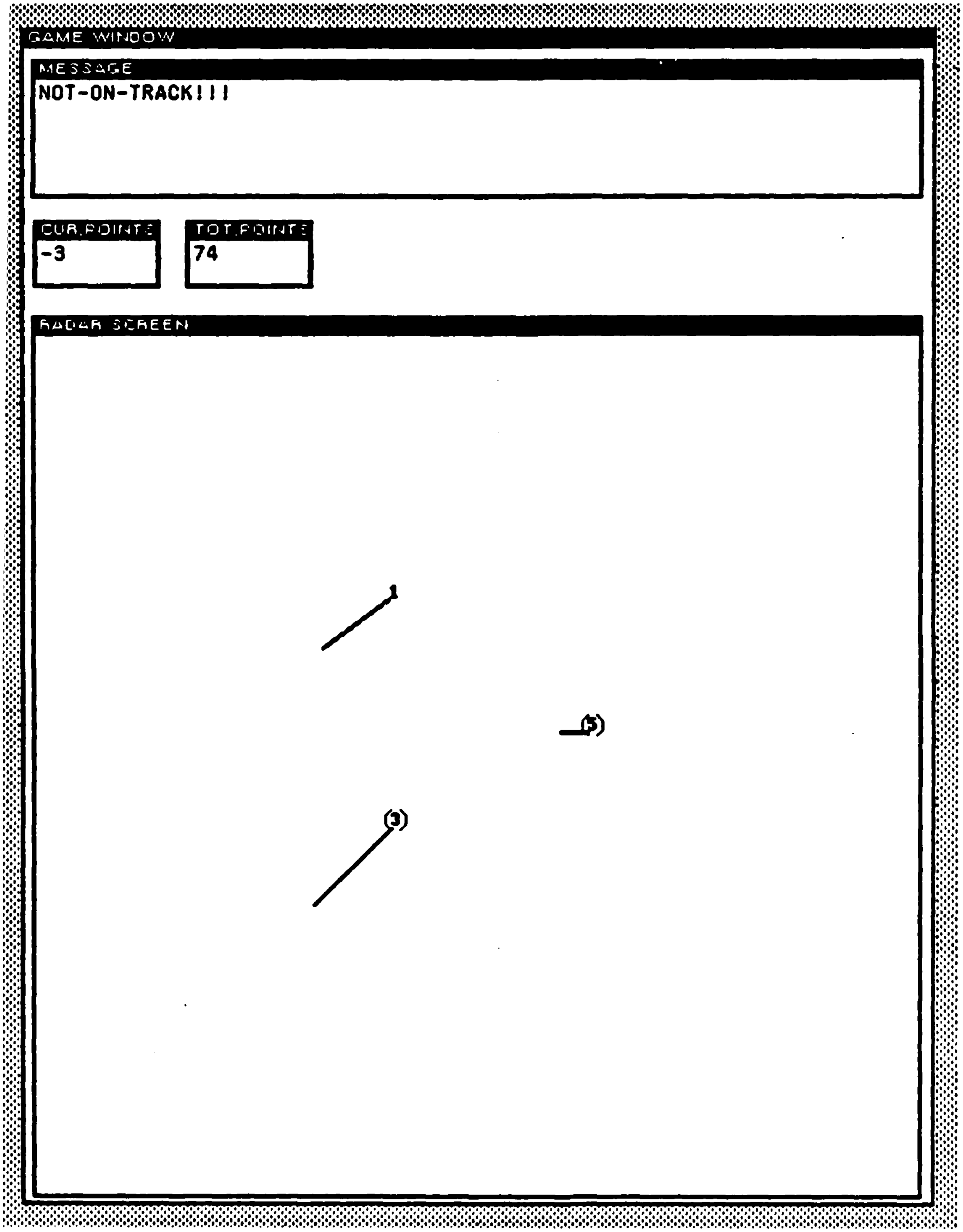
A student representation system that depends on introspective analyses of the meaning of student statements would provide an unreliable set of representations. While some student comments correlate well with observed performance behaviors, most behavior is not well reflected in the descriptions. What is required is a system that builds representations of student knowledge based on observed student behavior.

Building such representations as a dynamic task is carried out is one of goals of the current research effort. Two largely autonomous processes reside in a single computer system that both provides a task environment and represents the student's understanding of the task. The AIC hooking task was rewritten in Interlisp-D on a Xerox 1108 Scientific Information Processor. A system for representing aspects of knowledge about the task has been implemented, based on a set of generic schemata for dynamic skill primitives. When these schemata are activated during an interactive task session, they build a semantic network that represents the events of the session.

The AIC hooking task on the Xerox 1108 is implemented as a process bound to a set of windows that contain the simulated radar screen, scoring information, and performance feedback (see Figure 7). The blips change location at periodic intervals that represent the results of new radar sweeps.

As events occur in the simulation, a second process evaluates those events — the simulation events and the student responses — in terms of a pre-defined "vocabulary" of conceptual representations. The elements of this semantic vocabulary are LISP functions that play the role of schemata or frames. The schemata are activated by events in the task simulation and apply tests to events to determine whether they apply. When a schema finds that it does apply, it creates an instance node in a semantic network that represents an instantiation of that concept.

The representation of student knowledge about the AIC hooking task takes two forms in this system. Generic knowledge is represented by schemata written in LISP. Specific knowledge is represented by a semantic network created by and linked to the generic schemata. When a simulated radar sweep takes place, a number of generic knowledge elements are activated. For example, LOCATED-AT is activated for each object on the radar screen. This creates a number of new instances of LOCATED-AT in the network. These instantiations of LOCATED-AT represent the user's knowledge about the locations of objects at the end of a



NOT-ON-TRACK!!!

CUR.POINTS
-3

TOT.POINTS
74

Figure 7.

radar sweep. In a sense, the activations of LOCATED-AT serve as a functional model of the act of perceiving the locations of the objects.

Other, more top-down, schemata are also activated when a radar sweep takes place, such as a search for instances of UNLABELED blips on the screen. (Some students follow the strategy of noticing which blips are unlabeled after a radar sweep, and then rehook them.) UNLABELED is a complex schema that activates other schemata, such as NEAR and ANY. Thus, the search for unlabeled blips may result, not only in the addition of instantiations of UNLABELED to the representational network, but also in the instantiations of other schemata, such as NEAR.

Figure 8 presents a portion of a representational network immediately after a sweep has taken place. This figure represents a student's understanding of some of the spatial relationships that hold among objects on the screen. Node A0036, for example represents a memory for the fact that Blip1 is at a particular location on the screen (55, 473). It is an instantiation of the LOCATED-AT schema. Node A0034 is a representation of Blip1, an instantiation of the BLIP schema. And A0035 is the screen location, an instantiation of the LOCATION schema.

In Figure 8, the nodes are represented by ovals surrounding the node identification (a string such as A0036) and the name of the schema of which it is an instantiation. This use of the schema name in an oval is a graphic convention for a pointer labeled "instance" from the node to the generic schema. Another graphic abbreviation used in the figure is that each pointer shown actually represents two pointers, one the inverse of that displayed in the figure. Thus, an "object" pointer from A0036 to A0034 is matched, in the implementation of the representation, by an "object-of" pointer from A0034 to A0036.

The pointers of the representational network are implemented as properties. Each node has a set of properties with property values that are the nodes pointed to. Generic elements also have property lists.

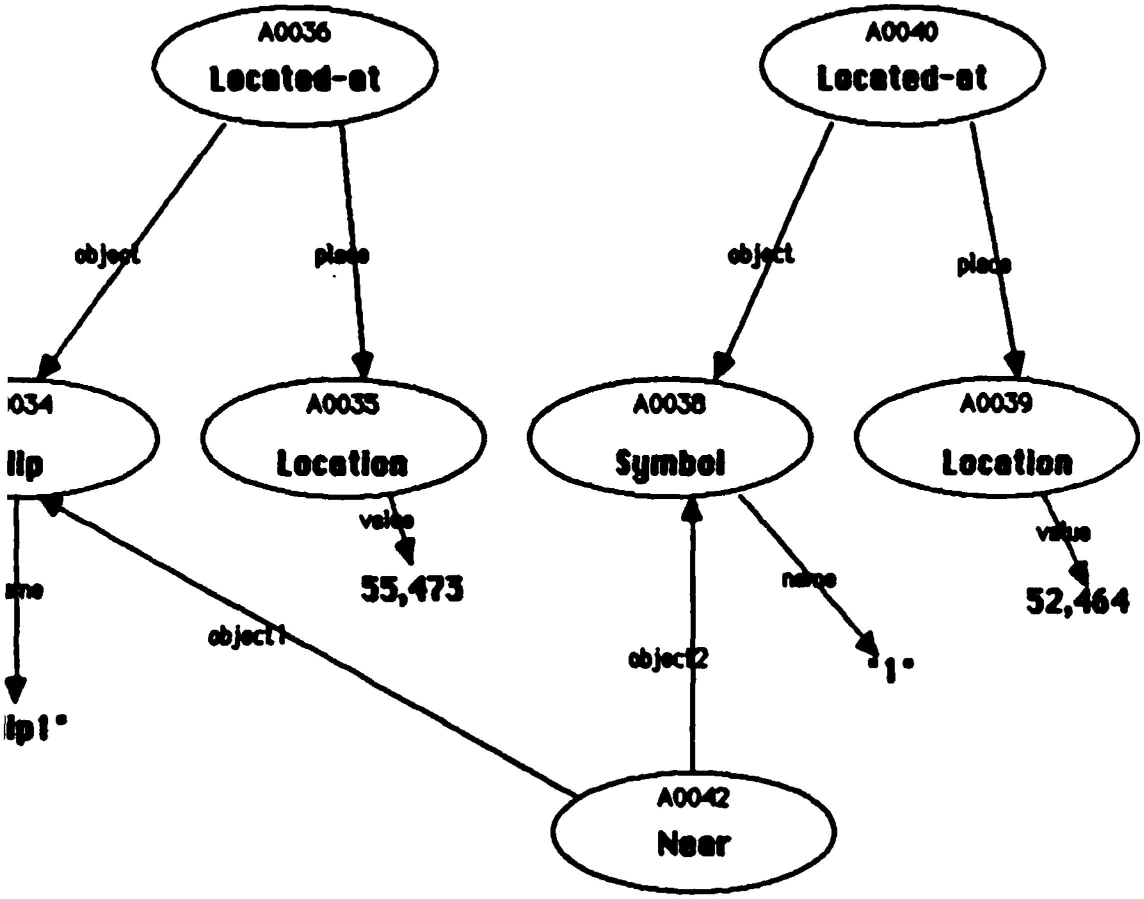

A0036
Located-at
object
place
A0040
Located-at
object
place
034
lip
A0035
Location
value
55,473
A0038
Symbol
name
"1"
A0039
Location
value
52,464
me
lip1"
object1
object2
A0042
Near

Figure 8.

"Instance" properties link generic schemata to the specific elements in the representational network.

Because the generic elements are LISP functions, a simple question answering facility is inherent in the representational system. The AIC simulation can be suspended at any point and a query window opened on the screen, as in Figure 10. The query window permits a user to enter standard schemata in a mode that returns a textual result. For example, the query

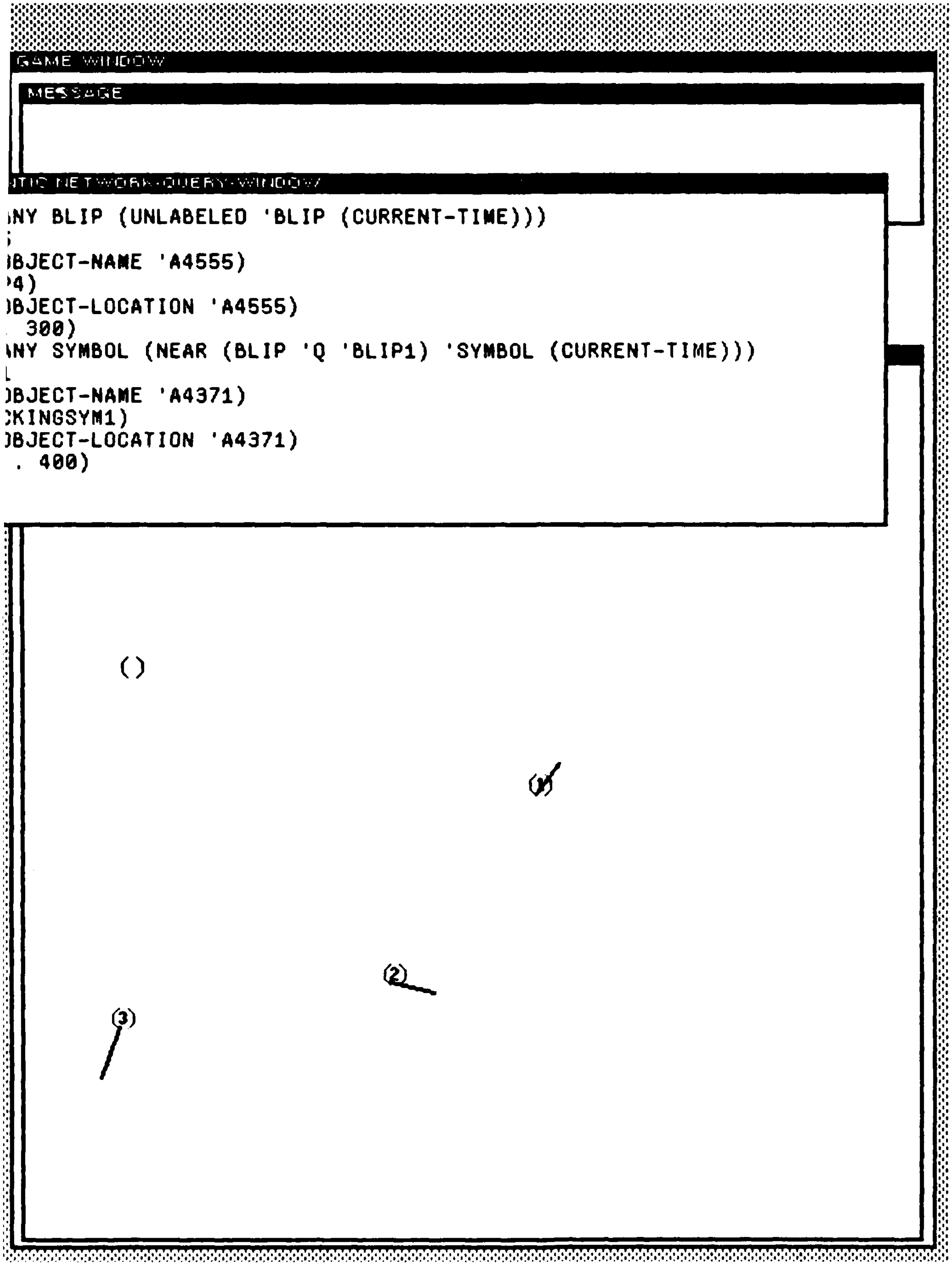
```
     (NEAR (BLIP BLIP1) (SYMBOL 1) CURRENT-TIME)
```

will return the node (A0042) that represents the perception of the nearness of Blip 1 and Symbol 1. (If Blip 1 and Symbol 1 were not in fact near each other, then the query would return NIL.) In Figure 9, a query window has been opened, suspending the AIC simulation task.

Directions for Future Development

The work done thus far has laid the foundations for a system that automatically constructs models of student understanding of dynamic skill tasks. Such a system would work in conjunction with a task simulator that would provide students with training practice. Observational evidence suggests that dynamic skill task knowledge is quite variable. An effective intelligent trainer would benefit from the availability of an accurate model of student knowledge, built on the basis of student interactions.

Such an automatic knowledge representation system would require a large library of generic schemata representing the different ways that task performers can view or understand aspects of the task domain. For example, there might be several different schemata for NEAR, representing different performers' conceptions of how close two elements must be for them to be considered near to each other. As a session progresses, evidence (in the form of performer responses to simulated events) will accumulate in support of certain schemata and against others. The set of supported schemata will comprise a model of the



```
NY BLIP (UNLABELED 'BLIP (CURRENT-TIME)))
;
BJECT-NAME 'A4555)
4)
BJECT-LOCATION 'A4555)
 300)
NY SYMBOL (NEAR (BLIP 'Q 'BLIP1) 'SYMBOL (CURRENT-TIME)))
L
BJECT-NAME 'A4371)
CKINGSYM1)
BJECT-LOCATION 'A4371)
. 400)
```

Figure 9.

student's understanding of the task.

Once such a student model has been created, it can be used to guide tutorial interactions. The system will have access to an ideal model of task knowledge, which can be compared with the current student model. It remains to be determined what are the most effective methods for producing tutorials using the current student model and the ideal model. An interesting possible method would be to demonstrate the differences in the performance of the task to a student by simulating the performance of an expert, using the ideal model, comparing that performacne with that of the student model. It is even conceivable that some simple verbal explanations could be generated by processes that compare the ideal and the student schemata.

REFERENCES

Allen, J. F. Towards a general theory of action and time. Artificial Intelligence, 1984, 23, 123-154.

Boring, A. Thinglab -- A constraint-oriented simulation laboratory. Palo Alto: Xerox PARC Technical Report SSL-79-3, 1979.

Fillmore, C. J. The case for case. In E. Bach & R. Harms (Eds.), Universals in linguistic theory. New York: Holt, Rinehart, ' Winston, 1-88.

Hollan, J. D. STEAMER: An overview tih implications for AI applications in other domains. Presented at the Joint Services Workshop on Artificial Intelligence in Maintenance: Automated Testing, Maintenance Aiding, and Training, Boulder, CO: Institute of Cognitive Science, October, 1983.

Kieras, D. E. A simulation model for procedure inference from a mental model for a simple device. Tucson, Arizona: University of Arizona, Department of Psychology, Technical Report No. 15, May 1984.

Kieras, D. E. & Bovair, S. The role of a mental model in learning to operate a device. Tucson, Arizona: University of Arizona, Department of Psychology, Technical Report No. 15, May 1984.

Kintsch, W. The representation of meaning in memory. Hillsdale, N. J.: Lawrence Erlbaum Associates, 1974.

McCarthy, J. & Hayes, P. J. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (Eds.) Machine Intelligence 4. Edinburgh: Edinburgh University Press, 1969.

Munro, A. Object oriented construction systems in Modula-2. Professional Program Record -- Wescon/84, November 1984, 26/3, 1-6.

Munro, A., Fehling, M. R. Blais, P., & Towne, D. M. Intrusive and non-intrusive instruction in dynamic skill training. Los Angeles: Behavioral Technology Laboratories, University of Southern California, Technical Report No. 97, December 1984.

Munro, A., Fehling, & Towne, D. M. Instruction intrusiveness in dynamic skill training. Journal of Computer Based Instruction, in press.

Munro, A., Towne, D. M. & Fehling, M. R. An experimental system for research on dynamic skills training. Los Angeles: Behavioral Technology Laboratories, University of Southern California, Technical Report No. 96, December 1984.

Munro, A., Towne, D. M., Cody, J. A. & Abramowski, H. Techniques for comptuer-based training of air intercept decision making skills. Los Angeles: Behavioral Technology Laboratories, University of Southern California, Technical Report No. 101, December 1984.

Schneiderman, B. The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology, 1982, 1, 237-256.

Towne, D. A. & Munro, A. Preliminary design of the advanced Equipment Simulation Authoring System. Los Angeles: Behavioral Technology Laboratories, University of Southern California, Technical Report No. 105, December 1984.

Navy

1 Robert Ahlers
Code N711
Human Factors Laboratory
NAVTRAEQUIPCEN
Orlando, FL 32813

1 Dr. Ed Aiken
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Meryl S. Baker
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Robert Breaux
NAVTRAEQUIPCEN
Code N-095R
Orlando, FL 32813

1 Dr. Richard Cantone
Navy Research Laboratory
Code 7510
Washington, DC 20375

1 Dr. Robert Carroll
NAVOP 115
Washington , DC 20370

1 Dr. Susan Chipman
Code 442PT
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22217

1 Dr. Stanley Collyer
Office of Naval Technology
800 N. Quincy Street
Arlington, VA 22217

1 CDR Mike Curran
Office of Naval Research
800 N. Quincy St.
Code 270
Arlington, VA 22217

1 Dr. Charles E. Davis
Personnel and Training Research
Office of Naval Research (Code 442PT)
800 North Quincy Street
Arlington, VA 22217

Navy

1 Edward E. Eddowes
CNATRA N301
Naval Air Station
Corpus Christi, TX 78419

1 Dr. Marshall J. Farr
2520 North Vernon Street
Arlington, VA 22207

1 Dr. Jude Franklin
Code 7510
Navy Research Laboratory
Washington, DC 20375

1 Dr. Jim Hollan
Code 51
Navy Personnel R & D Center
San Diego, CA 92152

1 Dr. Ed Hutchins
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Norman J. Kerr
Chief of Naval Education and Training
Code 0CA2
Naval Air Station
Pensacola, FL 32508

1 Dr. Peter Kincaid
Training Analysis & Evaluation Group
Dept. of the Navy
Orlando, FL 32813

1 Dr. William L. Maloy (02)
Chief of Naval Education and Training
Naval Air Station
Pensacola, FL 32508

1 Dr. Joe McLachlan
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. James McMichael
Navy PersonnelR&D Center
San Diego, CA 92152

1 Dr William Montague
NPRDC Code 13
San Diego, CA 92152

1 Technical Director
Navy Personnel R&D Center
San Diego, CA 92152

Navy

1 Office of Naval Research
Code 433
800 N. Quincy SStreet
Arlington, VA 22217

1 Psychological Sciences Division
Code 442
Office of Naval Research
Arlington, VA 22217

1 Director
Engineering Psychology Program
Code 442EP
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217

1 Organizational Effectiveness
Research Program, Code 442OE
Office of Naval Research
Arlington, VA 22217

1 Psychologist
ONR Branch Office
1030 East Green Street
Pasadena, CA 91101

1 Dr. Gil Ricard
Code N711
NTEC
Orlando, FL 32813

1 William Rizzo
Code 712 NTEC
Orlando, FL 32813

1 Dr. Paul B. Schneck
Office of Naval Research
Code 433
800 N. Quincy
Arlington, VA 22217

1 Dr. Michael G. Shafto
ONR Code 442PT
800 N. Quincy Street
Arlington, VA 22217

1 Dr. Alfred F. Smode
Senior Scientist
Code 7B
Naval Training Equipment Center
Orlando, FL 32813

Navy

1 Dr. Richard Sorensen
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Thomas Sticht
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Martin A. Tolcott
Leader, Psychological Sciences Division
Office of Naval Research
800 N. Quincy St.
Arlinsgon, VA 22217

1 Dr. James Tweeddale
Technical Director
Navy Personnel R&D Center
San Diego, CA 92152

1 Roger Weissinger-Baylon
Department of Administrative Sciences
Naval Postgraduate School
Monterey, CA 93940

1 Dr. Douglas Wetzel
Code 12
Navy Personnal R&D Center
San Diego, CA 92152

1 Dr. Wallace Wulfeck, III
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Steven Zornetzer
Associate Director for Life Sciences
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22217

Army

1 Technical Director
U. S. Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Dr. Beatrice J. Farr
U. S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Dr. Milton S. Katz
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Dr. Harold F. O'Neil, Jr.
Director, Training Research Lab
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Commander, U.S. Army Research Institute
for the Behavioral & Social Sciences
ATTN: PERI-BR (Dr. Judith Orasanu)
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Dr. Ray Perez
5001 Eisenhower Avenue
PERI-II
Alexandria, VA 22333

1 Joseph Psotka, Ph.D.
ATTN: PERI-1C
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333

1 Dr. Robert Sasmor
U. S. Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

1 DR. ROBERT J. SEIDEL
US Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333

Army

1 Dr. Zita M. Simutis
Chief, Instructional Technology
Systems Area
ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Air Force

1 U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, DC 20332

1 Dr. William E. Alley
AFHRL/MOT
Brooks AFB
, TX 78235

1 Dr. Earl A. Alluisi
HQ, AFHRL (AFSC)
Brooks AFB, TX 78235

1 Bryan Dallman
AFHRL/LRT
Lowry AFB, CO 80230

3 Dr. Sherrie Gott
AFHRL/MODJ
Brooks AFB
, TX 78235

1 Dr. Patrick Kyllonen
AFHRL/MOE

Brooks AFB, TX 78235

1 Mr. Bill Neale
HQ ATC/TTA
Randolph AFB
, TX 78148

1 Dr. Lawrence E. Reed
Research Psychologist
AFHRL/LRG
Wright-
Patterson AFB
, OH 45433

1 Dr. John Tangney
AFOSR/NL
Bolling AFB, DC 20332

1 Dr. Joseph Yasatuke
AFHRL/LRT
Lowry AFB, CO 80230

Department of Defense

1 Dr. Craig I. Fields
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

1 Military Assistant for Training and
Personnel Technology
Office of the Under Secretary of Defens
for Research & Engineering
Room 3D129, The Pentagon
Washington, DC 20301

1 Major Jack Thorpe
DARPA
1400 Wilson Blvd.
Arlington, VA 22209

1 Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria , VA 22333

Civilian Agencies

1 Dr. Patricia A. Butler
NIE-BRN Bldg, Stop # 7
1200 19th St., NW
Washington, DC 20208

1 Mr. Jim Carey
Coast Guard G-PTE
2100 Second St., S.W.
Washington, DC 20593

1 Dr. Arthur Melmed
724 Brown
U. S. Dept. of Education
Washington, DC 20208

1 Dr. Andrew R. Molnar
Office of Scientific and Engineering
Personnel and Education
National Science Foundation
Washington, DC 20550

1 Dr. Douglas Pearse
DCIEM
Box 2000
Downsview, Ontario
CANADA

1 Dr. Judy Segal
NIE
1200 19th Street N.W.
Mail Stop 6
Washington, DC 20208

1 Dr. Ramsay W. Selden
National Institute of Education
1200 19th St., NW
Washington, DC 20208

1 Dr. Sylvia A. S. Shafto
National Institute of Education
1200 19th Street
Mail Stop 6
Washington, DC 20208

1 Dr. Frederick Steinheiser
CIA-ORD
612 Ames
Washington, DC 20505

1 Dr. Joseph L. Young, Director
Memory & Cognitive Processes
National Science Foundation
Washington, DC 20550

Private Sector

1 Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

1 Patricia Baggett
Department of Psychology
University of Colorado
Boulder, CO 80309

1 Eva L. Baker
Director
UCLA Center for the Study of Evaluation
145 Moore Hall
University of California, Los Angeles
Los Angeles, CA 90024

1 Mr. Avron Barr
Department of Computer Science
Stanford University
Stanford, CA 94305

1 Dr. Isaac Bejar
Educational Testing Service
Princeton, NJ 08450

1 Gordon H. Bower
Department of Psychology
Stanford University
Stanford, CA 94306

1 Dr. John S. Brown
XEROX Palo Alto Research Center
3333 Coyote Road
Palo Alto, CA 94304

1 Dr. Glenn Bryan
6208 Poe Road
Bethesda, MD 20817

1 Dr. Bruce Buchanan
Department of Computer Science
Stanford University
Stanford, CA 94305

1 Dr. Jaime Carbonell
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

1 Dr. Pat Carpenter
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Private Sector

1 Dr. Micheline Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

1 Dr. William Clancey
Department of Computer Science
Stanford University
Stanford, CA 94306

1 Dr. Michael Cole
University of California
at San Diego
Laboratory of Comparative
Human Cognition - D003A
La Jolla, CA 92093

1 Dr. Allan M. Collins
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

1 Dr. Lynn A. Cooper
LRDC
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

1 Lee Cronbach
16 Laburnum Road
Atherton, CA 94205

1 Dr. Jeffrey Elman
University of California, San Diego
Department of Linguistics
La Jolla, CA 92093

1 Dr. Anders Ericsson
Department of Psychology
University of Colorado
Boulder, CO 80309

1 Dr. Paul Feltovich
Department of Medical Education
Southern Illinois University
School of Medicine
P.O. Box 3926
Springfield, IL 62708

Private Sector

1 Mr. Wallace Feurzeig
Department of Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

1 Dr. Dexter Fletcher
University of Oregon
Department of Computer Science
Eugene, OR 97403

1 Dr. John R. Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge, MA 02138

1 Dr. R. Edward Geiselman
Department of Psychology
University of California
Los Angeles, CA 90024

1 Dr. Michael Genesereth
Department of Computer Science
Stanford University
Stanford, CA 94305

1 Dr. Dedre Gentner
Psychology Department
University of Illinois
603 East Daniel Street
Champaign, IL 61820

1 Dr. Don Gentner
Center for Human Information Processing
University of California, San Diego
La Jolla, CA 92093

1 Dr. Robert Glaser
Learning Research & Development Center
University of Pittsburgh
3939 O'Hara Street
PITTSBURGH, PA 15260

1 Dr. Josph Goguen
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

1 Dr. Daniel Gopher
Faculty of Industrial Engineering
& Management
TECHNION
Haifa 32000
ISRAEL

Private Sector

1 Dr. Bert Green
Johns Hopkins University
Department of Psychology
Charles & 34th Street
Baltimore, MD 21218

1 Dr. James G. Greeno
University of California, Berkeley
Department of Education

Berkeley , CA

1 Dr. Henry M. Halff
Halff Resources
4918 33rd Road, North
Arlington, VA 22207

1 Dr. Barbara Hayes-Roth
Department of Computer Science
Stanford University
Stanford, CA 95305

1 Dr. Joan I. Heller
Graduate Group in Science and
Mathematics Education
c/o School of Education
University of California
Berkeley, CA 94720

1 Dr. Earl Hunt
Dept. of Psychology
University of Washington
Seattle, WA 98105

1 Dr. Zachary Jacobson
Bureau of Management Consulting
365 Laurier Avenue West
Ottawa, Ontario K1A 0S5
CANADA

1 Dr. Marcel Just
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

1 Dr. Demetrios Karis
Department of Psychology
University of Illinois
603 E. Daniel Street
Champaign, IL 61820

Private Sector

1 Dr. Scott Kelso
Haskins Laboratories, Inc
270 Crown Street
New Haven, CT 06510

1 Dr. David Kieras
Program in Technical Communication
College of Engineering
1223 E. Engineering Building
University of Michigan
Ann Arbor, MI 48109

1 Dr. Janet L. Kolodner
Georgia Institute of Technology
School of Information &
Computer Science
Atlanta, GA 30332

1 Dr. Stephen Kosslyn
1236 William James Hall
33 Kirkland St.
Cambridge, MA 02138

1 Dr. Pat Langley
University of California, Irvine
Department of Information and
Computer Science
Irvine , CA 92717

1 Dr. Jill Larkin
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213

1 Douglas Lenat
Computer Science Department
Stanford University
Stanford, CA 94306

1 Dr. Alan Lesgold
Learning R&D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

1 Dr. Jim Levin
University of California
at San Diego
Laboratory fof Comparative
Human Cognition - D003A
La Jolla, CA 92093

Private Sector

1 Dr. Michael Levine
Department of Educational Psychology
210 Education Bldg.
University of Illinois
Champaign, IL 61801

1 Dr. Marcia C. Linn
Lawrence Hall of Science
University of California
Berkeley, CA 94720

1 Dr. Don Lyon
P. O. Box 44
Higley , AZ 85236

1 Dr. Jay McClelland
Department of Psychology
MIT
Cambridge, MA 02139

1 Dr. Kathleen McKeoun
Columbia University
Department of Computer Science
New York, NY 10027

1 Dr. Mark Miller
Computer*Thought Corporation
1721 West Plano Parkway
Plano, TX 75075

1 Dr. Allen Munro
Behavioral Technology Laboratories
1845 Elena Ave., Fourth Floor
Redondo Beach, CA 90277

1 Dr. Donald A Norman
Cognitive Science, C-015
Univ. of California, San Diego
La Jolla, CA 92093

1 Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311

1 Dr. James W. Pellegrino
University of California,
Santa Barbara
Dept. of Psychology
Santa Barabara , CA 93106

Private Sector

1 Dr. Tjeerd Plomp
Twente University of Technology
Dept. of Education
7500 AE ENSCHEDE
P.O. Box 217
THE NETHERLANDS

1 Dr. Steven E. Poltrock
MCC
9430 Research Blvd.
Echelon Bldg #1
Austin , TX 78759

1 Dr. Mike Posner
Department of Psychology
University of Oregon
Eugene, OR 97403

1 Dr. James A. Reggia
University of Maryland
School of Medicine and Hospital
Departrent of Neurology
22 South Greene Street
Baltimore, MD 21201

1 Dr. Fred Reif
Physics Department
University of California
Berkeley, CA 94720

1 Dr. Lauren Resnick
LRDC
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 1521

1 Dr. Jeff Richardson
Denver Research Institute
University of Denver
Denver, CO 80208

1 Mary S. Riley
Program in Cognitive Science
Center for Human Information Processing
University of California, San Diego
La Jolla, CA 92093

1 Dr. Ernst Z. Rothkopf
Bell Laboratories
Murray Hill, NJ 07974

Private Sector

1 Dr. William B. Rouse
Georgia Institute of Technology
School of Industrial & Systems
Engineering
Atlanta, GA 30332

1 Dr. David Rumelhart
Center for Human Information Processing
Univ. of California, San Diego
La Jolla, CA 92093

1 PROF. FUMIKO SAMEJIMA
DEPT. OF PSYCHOLOGY
UNIVERSITY OF TENNESSEE
KNOXVILLE, TN 37916

1 Dr. Michael J. Samet
Perceptronics, Inc
6271 Variel Avenue
Woodland Hills, CA 91364

1 Dr. Arthur Samuel
Yale University
Department of Psychology
Box 11A, Yale Station
New Haven, CT 06520

1 Dr. Walter Schneider
Psychology Department
603 E. Daniel
Champaign, IL 61820

1 Dr. Alan Schoenfeld
University of California, Berkeley
Department of Education
Berkeley , CA

1 Dr. H. Wallace Sinaiko
Program Director
Manpower Research and Advisory Services
Smithsonian Institution
801 North Pitt Street
Alexandria, VA 22314

1 Dr. Eliott Soloway
Yale University
Department of Computer Science
P.O. Box 2158
New Haven, CT 06520

1 Dr. Kathryn T. Spoehr
Psychology Department
Brown University
Providence, RI 02912

Private Sector

1 James J. Staszewski
Research Associate
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

1 Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

1 DR. PATRICK SUPPES
INSTITUTE FOR MATHEMATICAL STUDIES IN
THE SOCIAL SCIENCES
STANFORD UNIVERSITY
STANFORD, CA 94305

1 Dr. Martin M. Taylor
DCIEM
Box 2000
Downsview, Ontario
CANADA

1 Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1185 Coleman Avenue, Box 580
Santa Clara, CA 95052

1 Dr. Douglas Towne
Univ. of So. California
Behavioral Technology Labs
1845 S. Elena Ave.
Redondo Beach, CA 90277

1 Dr. Kurt Van Lehn
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

1 Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1185 Coleman Ave., Box 580
Santa Clara, CA 95052

1 Dr. Mike Williams
IntelliGenetics
124 University Avenue
Palo Alto, CA 94301

Private Sector

1 Mr. Carl York
System Development Foundation
181 Lytton Avenue, Suite 210
Palo Alto, CA 94301